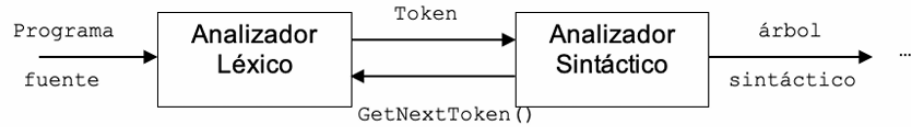


Verifica que los tokens formen una estructura gramaticalmente correcta.

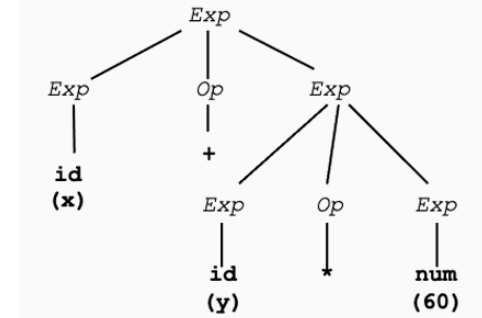


3.1. ANÁLISIS SINTÁCTICO

3.2. GRAMÁTICAS LIBRES DE CONTEXTO

Conjunto de reglas formales para definir la sintaxis de un lenguaje.

Árbol sintáctico



Usa una pila y una tabla para guiar el análisis de forma iterativa.

Movimientos que realiza un analizador sintáctico predictivo con la entrada id * id + id

Pila	Entrada	Acción
\$E	id*id-id\$	$E \rightarrow TE'$
SE'	id*id-id\$	$T \rightarrow FT'$
$SE'T'$	id*id-id\$	$F \rightarrow id$
$SE'T'id$	id*id-id\$	concuerta(id)
$SE'T'$	*id-id\$	$T' \rightarrow *FT'$
$SE'T'F'$	*id-id\$	concuerta(*)
$SE'T'F$	id-id\$	$F \rightarrow id$
$SE'T'id$	id-id\$	concuerta(id)
$SE'T'$	+id\$	$T' \rightarrow \epsilon$
SE'	+id\$	$E' \rightarrow +TE'$
$SE'T+$	+id\$	concuerta(+)
$SE'T$	id\$	$T \rightarrow FT'$
$SE'T'F$	id\$	$F \rightarrow id$
$SE'T'id$	id\$	concuerta(id)
$SE'T'$	\$	$T' \rightarrow \epsilon$
SE'	\$	$E' \rightarrow \epsilon$
\$	\$	aceptar()

Tipo de gramática que permite un análisis predictivo sin ambigüedades.

Algoritmo 3.1: Construcción de una tabla de análisis sintáctico predictivo (Aho, Lam, Sethi, & Ullman, 2008, p. 224).

Entrada: La gramática G .

Salida: La tabla de análisis sintáctico M .

Método: Para cada producción $A \rightarrow \alpha$ de la gramática, hacer lo siguiente:

- Para cada terminal a en $\text{Primero}(\alpha)$, agregar $A \rightarrow \alpha$ a $M[A, a]$.
- Si ϵ está en $\text{Primero}(\alpha)$, entonces para cada terminal b en $\text{Siguiente}(A)$, se agrega $A \rightarrow \alpha$ a $M[A, b]$. Si ϵ está en $\text{Primero}(\alpha)$ y $\$$ se encuentra en $\text{Siguiente}(A)$, se agrega $A \rightarrow \alpha$ a $M[A, \$]$ también.

3.8. ANÁLISIS SINTÁCTICO PREDICTIVO NO RECURSIVO

3.7. GRAMÁTICAS LL(1)

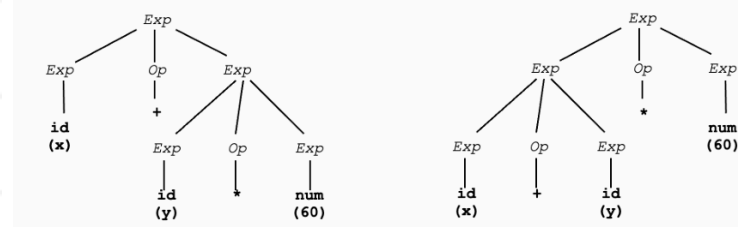
3.6. CONJUNTOS PRIMERO Y SIGUIENTE

3.5. ANÁLISIS SINTÁCTICO DESCENDENTE RECURSIVO

3.3. ÁRBOLES DE ANÁLISIS SINTÁCTICO

Representación gráfica en forma de árbol de la estructura de una frase.

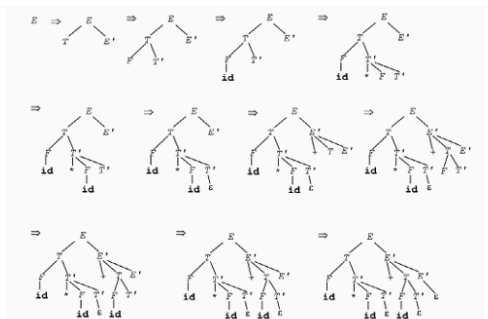
Dos árboles sintácticos distintos para la misma expresión



3.4. ANÁLISIS SINTÁCTICO DESCENDENTE

Construye el árbol de análisis desde la raíz hacia las hojas.

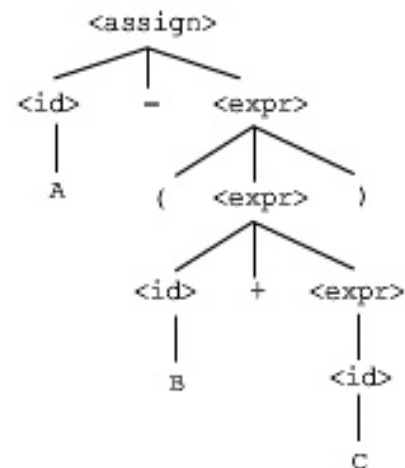
Análisis sintáctico descendente para id*id+id



Implementa el análisis descendente usando recursión en el código.

Bosquejo de un procedimiento para un no terminal en un analizador sintáctico descendente

```
void A() {
    Elegir una producción  $A, A \rightarrow X_1X_2...X_k$  ;
    for( i = 1 a k ){
        if(  $X_i$  es un no terminal )
            llamar al procedimiento  $X_i()$ ;
        else if(  $X_i$  es igual al símbolo de entrada actual a )
            avanzar la entrada hasta el siguiente símbolo;
        else /* rutina de error y retroceso() */;
    }
}
```



Parse Tree for the Statement $A = (B + C)$

Herramientas para predecir qué producción aplicar en el análisis.